Automated excavation and detection of Design Patterns

Szabolcs Marien

The resent specification languages can formalize the patterns using additional language elements like: graphical, formalization tool, but there is no support for detection of Design Patterns in the design phase and excavation of Design Patterns.

If the automatic detection of Design Patterns were solved in the design phase, then the cost end effort of development could be significantly reduced, since the design failures can be eliminated during the design phase.

The excavation of Design Patterns is very hard. There is no solution to do it easier at the moment. The reason of it is that the representing forms of Design Patterns are not applicable for building on them a proofing methodology.

On the one hand, we set up a simple set based language with which we can formulate the Design Patterns as existential logical forms. To be more precise, we formalize the first part: the problem specification part of patterns. The second part: the problem solution, is usually overlooked in the literature. We try to formalize the problem-solution, too. We understand the problem- solution as a constructive proof.

On the other hand, we show the recognized general roles of the type of creational Design Patterns, which can be used for realization the mapping between the UML class diagram and the formulae (mapping theorem). Using the mapping between the UML class diagrams of the design and formulae of the design components and using the logical formulae of Design Patterns, we try to show the automatic detection facilities of Design Patterns.

On the third hand, we show how to realize the automated excavation of Design Patterns using the previously mentioned mapping theorem and the constructive proofing technique.